symmetry

MDPI

Article

Security and Privacy Analysis of Vinoth et al.'s Authenticated Key Agreement Scheme for Industrial IoT

Da-Zhi Sun

Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing, Tianjin University, Tianjin 300350, China; sundazhi@tju.edu.cn; Tel.: +86-22-2740-1091

Abstract: Vinoth et al. proposed an authenticated key agreement scheme for industrial IoT (Internet of Things) applications. Vinoth et al.'s scheme aimed to protect the remote sensing data of industrial IoT devices under hostile environments. The scheme is interesting because the authorized user is allowed simultaneously to access the multiple IoT sensing devices. Therefore, we carefully analyzed the security and privacy implications of Vinoth et al.'s scheme. Our findings are summarized as follows. One, Vinoth et al.'s scheme failed to defeat user impersonation attacks. Second, Vinoth et al.'s scheme did not prevent IoT sensing device impersonation attacks. Third, Vinoth et al.'s scheme suffered from replay attacks. Fourth, Vinoth et al.'s scheme was vulnerable to desynchronization attacks. Fifth, Vinoth et al.'s scheme could not maintain user privacy. As a case study, our analysis results enlighten researchers and engineers on the design of robust and efficient authenticated key agreement schemes for IoT applications.

Keywords: IoT; sensing networks; authentication; key agreement; impersonation attack; replay attack; desynchronization attack; user privacy

Citation: Sun, D.-Z. Security and Privacy Analysis of Vinoth et al.'s Authenticated Key Agreement Scheme for Industrial IoT. *Symmetry* **2021**, *13*, 1952. https://doi.org/10.3390/sym13101952

Academic Editors: Weizhi Meng, Georgios Kambourakis and Jun Shao

Received: 15 September 2021
Accepted: 12 October 2021
Published: 16 October 2021

1. Introduction

The Internet of Things (IoT) is a fast development in the long and continuing revolution of communications and computing. The IoT has expanded the interconnection of billions of industrial and personal objects through IoT sensing devices, which are typically composed of sensors, actuators, microcontrollers, transceivers, and batteries. IoT sensing devices bound to objects deliver sensor information, act on their environments, and in some cases adapt for the overall management of a larger system, such as a factory [1] or a city [2]. Moreover, these devices always communicate each other and form a remote sensing network. As a typical scenario, Industrial IoT is deployed for achieving intelligent manufacturing because of its advantages in automatic monitoring and efficient control. Under the industrial IoT environment, sensing devices can be remotely accessed and controlled by authorized users. During the process of industrial production, sensing devices collect real-time data. Users obtain this real-time data and then send control commands according to said data.

IoT sensing security [3] is perhaps the most complex and immature area of cybersecurity. The following characteristics hinder secure IoT sensing:

(1) *Very large attack surfaces*: There is a wide variety of points of vulnerability in IoT sensing systems and a large amount of data that may be compromised.

(2) *Widespread deployment*: There is ongoing, rapid deployment of IoT arrangements in commercial and industrial environments and, more importantly, in critical infrastructure environments. Most IoT sensing devices are remote and out of control. These deployments are attractive targets for security attacks.

(3) *Constrained device resources*: IoT sensing devices are typically constrained, with limited memory, processing power, and power supply.

(4) *Low cost*: IoT sensing devices are always manufactured, purchased, and deployed in the millions. This fact provides great incentive for manufacturers and customers to minimize the cost of these devices.

Motivation of This Paper. In the normal course of things, the user requires simultaneously access to multiple IoT sensing devices for a complex industrial task. Because of serious security and privacy threats, IoT sensing devices, especially remote devices, are required to support mutual authentication and secret key establishment with their users. The authenticated key agreement scheme provides authentication and key establishment services among users and multiple IoT sensing devices. We therefore analyzed the security and privacy of the authenticated key agreement scheme. Our research focused on not only outside but inside attackers, i.e., malicious users and corrupt IoT sensing devices.

1.1. Industrial IoT Sensing Model and Its Authenticated Key Agreement Scheme

In this section, we describe the sensing model and authenticated key agreement scheme studied in this paper.

1.1.1. Industrial IoT Sensing Model

The sensing model is depicted in Figure 1. There are three categories of entities, i.e., gateway nodes (GWNs), users, and industrial IoT sensing devices.

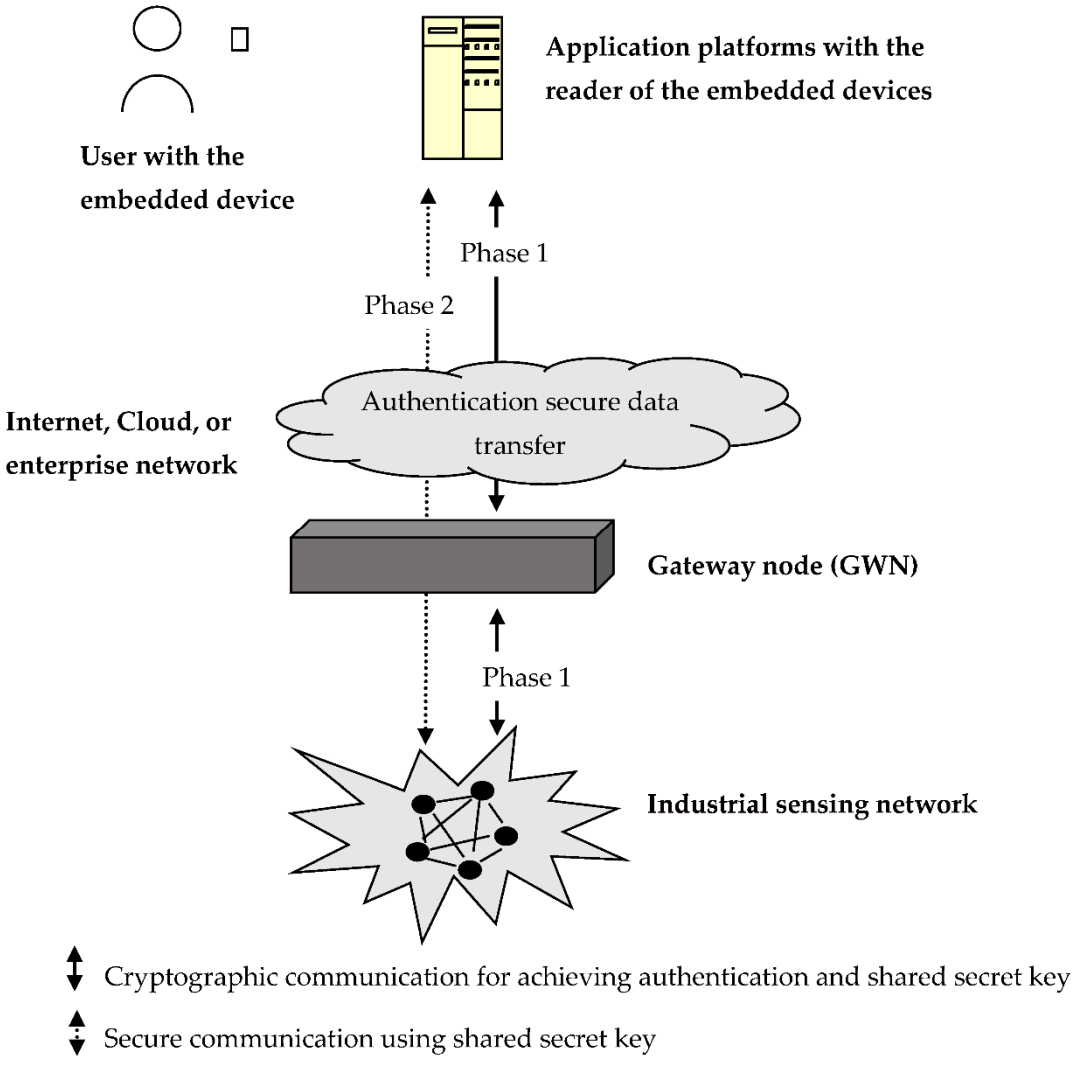

Figure 1. Industrial IoT sensing model.

(1) *GWN*: GWNs interconnect IoT sensing devices with high-level communication networks and perform the necessary translation between the protocols used in communication networks and those used in IoT sensing devices.

(2) *Users*: The users are allowed to access IoT sensing devices through GWNs. They gain security and privacy services with the help of embedded devices such as smart cards.

(3) *IoT sensing devices*: IoT sensing devices are utilized to monitor the status of objects and collect the information stored therein. Users can obtain the information collected by these devices in real time.

In our industrial IoT sensing model, we assumed that the users and the IoT sensing devices were untrusted entities. GWNs [4], meanwhile, cannot be compromised and were therefore considered to be fully trusted by the users and the IoT sensing devices. This assumption is reasonable because GWNs are usually placed in secure environments and equipped with tamper-resistant devices.

1.1.2. Authenticated Key Agreement Scheme

To set up a secure sensing network, the GWN initially writes some authentication credentials into IoT sensing devices. The user first registers to the GWN, and both the user and GWN write the authentication credentials into the user's embedded device. When the registered users want to access the deployed IoT sensing devices, they run an authentication session using their embedded device. During the authentication session, GWN helps the user and the IoT sensing devices authenticate each other and establish a shared secret key for subsequent secure communication. In addition, the users can change their authentication credentials, and the GWN can allow new IoT sensing devices to join the deployed sensing network and revoke existing devices from said network.

However, an attacker may exploit the vulnerabilities in the authenticated key agreement scheme to perform attacks, because the messages of the authentication session are often transmitted through a public channel, and this brings security problems in the industrial IoT environment. It is possible for an inside or outside attacker to impersonate an authorized user to obtain data by accessing sensing devices or to impersonate a legal IoT sensing device to provide fake data. These unsatisfactory security risks could lead to the destruction of industrial activity.

1.2. Related Work

In recent years, many authenticated key agreement schemes [5,6] have been proposed for IoT remote sensing environments, such as industrial IoT, telemedicine, and smart home. We review previous work on four dimensions.

From the user credentials perspective, authenticated key agreement schemes are classified into two categories, i.e., two- and three-factor (multifactor) schemes. In two-factor schemes [7–15], the security of the user is protected by both the secret key stored in the smart card and the human-memorizable password, and the user applies the password and the smart card to complete the authentication session. Compared to two-factor schemes, three-factor schemes [4,16–22] add biometrics to the user credentials; that is, the user must provide the smart card, the password, and biometrics at the same time.

In many privacy applications, the users does not want authentication sessions to be associated with their identity. This means that the user's identity is disclosed only to an authorized set of GWN and IoT sensing devices during the authentication sessions. Therefore, to preserve user privacy, authenticated key agreement schemes [23–27] thwart attempts to disclose or link users' identities by exploiting their authentication sessions.

Many researchers have extended authenticated key agreement schemes [28–34] to multi-gateway IoT environments. These revised schemes provide the user with a single sign under a set of GWNs. That is, when the user is authenticated by a GWN in the set of GWNs, he/she can access all IoT sensing devices governed by the set of GWNs even if the devices in question are not directly managed by the specific GWN that authenticated the

user. In addition, the multi-gateway schemes can solve the packet-collision problem due to single GWN mode.

Users often access multiple IoT sensing devices to complete complex tasks. It is inefficient for the user to run a separate authentication session with each IoT sensing device. Moreover, the logical relevance of the authentications and the shared secret keys cannot be guaranteed if the user independently runs several authentication sessions for a task. Hence, some authenticated key agreement schemes [4,35,36] have recently begun to provide authentication and group secret-key establishment between the user and multiple IoT sensing devices in an authentication session.

1.3. Our Contributions

In the IEEE Internet of Things Journal, Vinoth et al. [4] proposed an authenticated key agreement scheme that aimed to protect the remote sensing data of industrial IoT under the hostile environments. We carefully analyzed security and privacy under Vinoth et al.'s scheme. Our results are as follows.

(1) Vinoth et al.'s scheme failed to defeat a user impersonation attack. A legal but malicious user could impersonate IoT sensing devices, other users, and the GWN.

(2) Vinoth et al.'s scheme did not prevent IoT sensing device impersonation attacks. A legal but corrupt IoT sensing device can impersonate users, the GWN, and other IoT sensing devices.

(3) Vinoth et al.'s scheme suffered from replay attacks. Attackers can reuse the previous message in the authentication session to cheat the user and the GWN.

(4) Vinoth et al.'s scheme was vulnerable to desynchronization attacks. In these attacks, an attacker induces an inconsistent internal status between the user and the GWN. This security flaw causes the GWN to deny the service for the user.

(5) Vinoth et al.'s scheme cannot maintain user privacy. User identity is compromised during the run of the authentication session.

As a matter of convenience, in Table 1, we list some notation used throughout our paper.

Table 1. Description of notations.

Term	Definition
GWN, U	Gateway node and user
S_j	jth IoT sensing device
ID_{GWN}, ID_U, ID_{S_j}	GWN's, U's, and S_j's identities
TID_U	U's temporary identity for user anonymity
γ, K_{GWN}	GWN's long-term secret keys
$K_{GWN\text{-}U}$	Long-term secret key shared by GWN and U
PW	U's password
B, BK, τ	U's biometrics, biometrics key, and public reproduction parameter
s_j, f_j, k_j	S_j's secret parameters
$K_{GWN\text{-}S_j}$	Secret key shared by GWN and S_j
$K_{U\text{-}Sj}$	Secret session key shared by U and S_j
r_{GWN}, r_U, RN	Random numbers
TS_1, TS_2, TS_3, TS_4, TS_1', TS_2', TS_3', TS_4'	Timestamps
ΔTS	Maximum transmission delay
$\varphi()$	Vinoth et al.'s access structure function [4]
$Gen()/Rep()$	Generation algorithm/reproduction algorithm using biometrics fuzzy extractor
$h()$	Cryptographic hash function
$E_K()/D_K()$	Encryption algorithm/decryption algorithm using secret key K
mod	Congruent
$\oplus$, $\|$	Bitwise exclusive-or and concatenation

2. Review of Vinoth et al.'s Authenticated Key Agreement Scheme

2.1. Scheme Description

Vinoth et al.'s scheme is composed of seven phases: the offline sensing device registration phase, the user registration phase, the login phase, the authenticated key agreement phase, the biometrics and password update phase, the dynamically sensing device joining phase, and the sensing device revocation phase. For a self-contained discussion, we review the first four phases, which are related to our discussion. The full technical details of Vinoth et al.'s scheme can be found in [4].

2.1.1. Offline Sensing Device Registration Phase

GWN picks a unique ID_{S_j} for S_j, where $j = 1, 2, \ldots, n$. GWN then chooses a $K_{GWN\text{-}S_j}$ and two n-dimensional vectors $Vector_1$ and $Vector_2$ such that $K_{GWN\text{-}S_j} = Vector_1 \cdot x_0$ and $K_{GWN\text{-}S_j}{}^2 = Vector_2 \cdot x_0$, where $x_0 = \varphi(\text{GWN})$. GWN calculates $s_j = Vector_1 \cdot x_j$ and $f_j = Vector_2 \cdot x_j$, where $x_j = \varphi(S_j)$ $(1 \leq j \leq n)$. GWN computes and stores λ_t, where $K_{GWN\text{-}S_j} = \sum_{t=1}^{n} \lambda_t s_t$ and $K_{GWN\text{-}S_j}{}^2 = \sum_{t=1}^{n} \lambda_t f_t$. GWN selects the pairwise relative positive numbers k_j for each S_j $(1 \leq j \leq n)$. GWN computes $Mul = \prod_{j=1}^{n} k_j$ and $Mul_j = Mul/k_j$ and generates a random number $Nonce_j$ such that $Mul_j \times Nonce_j \equiv 1 \bmod k_j$. GWN computes $\gamma = \sum_{j=1}^{n} Var_j = \sum_{j=1}^{n} Mul_j \times Nonce_j$ and stores γ. GWN securely sends ID_{S_j}, s_j, f_j, and k_j to each S_j $(1 \leq j \leq n)$, and then S_j stores them. In the end, GWN deletes other messages.

2.1.2. User Registration Phase

Step 1: U chooses a unique ID_U and a PW, imprints the B, and computes $(BK, \tau) = Gen(B)$. U generates a random 128-bit number a, calculates $TPW = h(ID_U \| PW \| BK) \oplus a$, and securely sends the message $<ID_U, TPW>$ to GWN.

Step 2: When receiving $<ID_U, TPW>$, GWN randomly generates a 1024-bit K_{GWN} and a 128-bit TID_U, and then computes $K_{GWN\text{-}U} = h(ID_U \| K_{GWN})$, $A = K_{GWN\text{-}U} \oplus TPW$, and $C = ID_{GWN} \oplus TPW$. GWN stores TID_U, ID_U, and $K_{GWN\text{-}U}$ in its database. Finally, GWN writes $\{TID_U, A, C, h()\}$ into a smart card and securely sends the card to U.

Step 3: When receiving the card, U computes $RPW = h(ID_U \| PW \| BK)$, $A' = A \oplus TPW \oplus RPW$, $D = a \oplus h(ID_U \| BK)$, $C' = C \oplus TPW \oplus h(ID_U \| BK)$, and $V \equiv h(RPW \| A \| a \| h(ID_U \| BK)) \bmod \omega$, where ω is the medium integer [37–39]. Finally, U rewrites $\{TID_U, A', C', D, V, Gen(), Rep(), h(), \tau, \omega\}$ into the card.

2.1.3. Login Phase

Step 1: U inserts the smart card into the card reader, and then further inputs the ID_U and PW and imprints the B. The smart card reconstructs $BK = Rep(B, \tau)$ and computes $RPW = h(ID_U \| PW \| BK)$, $a = D \oplus h(ID_U \| BK)$, and $A = A' \oplus a$. The smart card further checks whether $V \equiv h(RPW \| A \| a \| h(ID_U \| BK)) \bmod \omega$. If not, the smart card terminates the login request.

Step 2: The smart card generates r_U and obtains current TS_1. The smart card computes $ID_{GWN} = C' \oplus h(ID_U \| BK)$, $M_1 = A' \oplus RPW \oplus r_U$, and $M_2 = h(TID_U \| M_1 \| ID_{GWN} \| r_U \| TS_1)$. In the end, the smart card sends the message $<TID_U, M_1, M_2, TS_1>$ to GWN.

2.1.4. Authenticated Key Agreement Phase

Step 1: After receiving $<TID_U, M_1, M_2, TS_1>$, GWN obtains the current TS_1' and checks the freshness of the login request by verifying whether $|TS_1 - TS_1'| \leq \Delta TS$. If not, GWN terminates this session. GWN searches its database by using the keyword TID_U and retrieves ID_U and $K_{GWN\text{-}U}$. GWN calculates $r_U = M_1 \oplus K_{GWN\text{-}U}$ and checks whether $M_2 = h(TID_U \| M_1 \| ID_{GWN} \| r_U \| TS_1)$. If not, GWN terminates this session. GWN generates r_{GWN} $(r_{GWN} \leq \min\{k_j\}, j = 1, 2, \ldots, n)$ and obtains the current TS_2. GWN computes $M_3 = r_{GWN} \times \gamma$, $M_4 = E_{rGWN}(ID_U, ID_{GWN}, r_U, r_{GWN} \oplus K_{GWN\text{-}U})$, and $M_5 = h(ID_U \| ID_{GWN} \| r_U \| M_3 \| K_{GWN\text{-}U} \| TS_2)$. Finally, GWN broadcasts the message $<M_3, M_4, M_5, TS_2>$ to all S_js.

Step 2: When receiving $<M_3, M_4, M_5, TS_2>$, each S_j obtains current TS_2' and checks the freshness of the message by verifying whether $|TS_2 - TS_2'| \leq \Delta TS$. If not, S_j terminates this session. S_j computes $r_{GWN} \equiv M_3 \bmod k_j$ and $(ID_U, ID_{GWN}, r_U, r_{GWN} \oplus K_{GWN\text{-}U}) = D_{rGWN}(M_4)$, and further checks whether $M_5 = h(ID_U \| ID_{GWN} \| r_U \| M_3 \| r_{GWN} \oplus K_{GWN\text{-}U} \oplus r_{GWN} \| TS_2)$. If not, S_j terminates this session. S_j computes $M_6 = E_{rGWN}(ID_{S_j}, s_j, f_j)$ and obtains current TS_3. S_j returns the message $<M_6, TS_3>$ to GWN.

Step 3: When receiving $<M_6, TS_3>$, GWN obtains current TS_3' and checks the freshness of the message by verifying whether $|TS_3 - TS_3'| \leq \Delta TS$. If not, GWN terminates the session. GWN computes $(ID_{S_j}, s_j, f_j) = D_{rGWN}(M_6)$ from each S_j. GWN calculates $\theta_1 = \sum_{t=1}^{n} \lambda_t s_t$ and $\theta_2 = \sum_{t=1}^{n} \lambda_t f_t$ and checks whether $\theta_1^2 = \theta_2$. If not, GWN terminates this session. GWN views θ_1 as $K_{GWN\text{-}S_j}$ and computes $M_7 = h(K_{GWN\text{-}S_j} \| r_{GWN})$, $M_8 = M_7 \times \gamma$, and $M_9 = h(M_7 \| M_8)$. Moreover, GWN generates a new temporary identity $TID_U{}^{new}$, obtains the current TS_4, and computes $M_{10} = E_{KGWN\text{-}U}(r_{GWN}, r_U, M_7)$, $M_{11} = h(ID_U \| K_{GWN\text{-}U} \| TS_4) \oplus TID_U{}^{new}$, and $M_{12} = h(M_{10} \| M_7 \| r_U)$. In the end, GWN broadcasts the message $<M_8, M_9>$ to all S_js and sends the message $<M_{10}, M_{11}, M_{12}, TS_4>$ to U.

Step 4: When receiving $<M_8, M_9>$, each S_j calculates $M_7 \equiv M_8 \bmod k_j$ and checks whether $M_9 = h(M_7 \| M_8)$. If not, S_j terminates this session. S_j computes $K_{U\text{-}Sj} = h(ID_U \| ID_{GWN} \| r_{GWN} \| r_U \| M_7 \| K_{GWN\text{-}U})$ and $M_{13} = h(K_{U\text{-}Sj} \| ID_{GWN} \| ID_U)$, and then sends the message $<M_{13}>$ to U.

Step 5: When receiving $<M_{10}, M_{11}, M_{12}, TS_4>$, U obtains the current TS_4' and checks the freshness of the message by verifying whether $|TS_4 - TS_4'| \leq \Delta TS$. If not, U terminates this session. U computes $(r_{GWN}, r_U, M_7) = D_{KGWN\text{-}U}(M_{10})$. Then, U checks whether the decrypted r_U is equal to the local r_U and $M_{12} = h(M_{10} \| M_7 \| r_U)$. If any one of them is unequal, U terminates this session. U calculates $K_{U\text{-}Sj} = h(ID_U \| ID_{GWN} \| r_{GWN} \| r_U \| M_7 \| K_{GWN\text{-}U})$. Furthermore, when receiving $<M_{13}>$ from each S_j, U checks whether $M_{13} = h(K_{U\text{-}Sj} \| ID_{GWN} \| ID_U)$. If not, U terminates this session. U computes $TID_U{}^{new} = h(ID_U \| K_{GWN\text{-}U} \| TS_4) \oplus M_{11}$ and replaces TID_U with $TID_U{}^{new}$.

Figure 2 shows the process of Vinoth et al.'s login phase and authenticated key agreement phase.

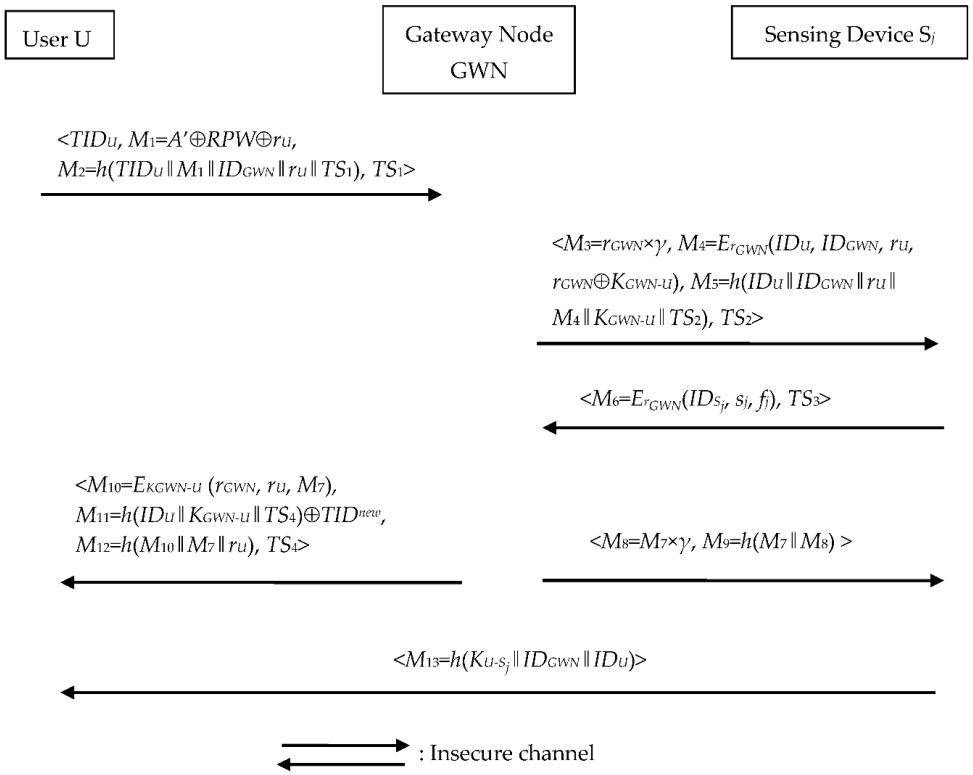

Figure 2. Vinoth et al.'s login phase and authenticated key agreement phase.

2.2. Vinoth et al.'s Security Assumption

Vinoth et al. claimed that their scheme was secure under the Canetti–Krawczyk threat model [40], which assumes that an attacker can eavesdrop on, intercept, modify, forge, and delete messages transmitted between any two entities over the public channel. An attacker can also impersonate users, IoT sensing devices, and the GWN to receive and send the messages. Furthermore, the attacker has the capability to expose some secrets of the users and the IoT sensing devices. More importantly, the attacker can be an insider, i.e., a user or an IoT sensing device, because users and sensing devices are untrusted entities. Under the Canetti–Krawczyk threat model, we discuss five types of attacks on Vinoth et al.'s scheme.

3. User Impersonation Attack

We showed that Vinoth et al.'s scheme was vulnerable to user impersonation attacks. That is, a legal but malicious user could impersonate IoT sensing devices, any other user, and the GWN in the deployed network. We assume that U_a is a legal but malicious user in Vinoth et al.'s scheme and maintains the identity ID_{Ua}, temporary identity TID_{Ua}, and long-term secret key $K_{GWN\text{-}Ua}$ shared with GWN.

3.1. Impersonation of IoT Sensing Devices

To impersonate a target S_j, U_a first initiates his/her authentication session with GWN. In Steps 2 and 3 of the authenticated key agreement phase, U_a eavesdrops on GWN's message $<M_6, TS_3>$ from S_j and S_j's $<M_8, M_9>$ from GWN. When U_a receives the message $<M_{10}, M_{11}, M_{12}, TS_4>$ in Step 3 of the authenticated key agreement phase, U_a computes $(r_{GWN}, r_U, M_7) = D_{KGWN\text{-}Ua}(M_{10})$. Now, U_a is able to compute $(ID_{S_j}, s_j, f_j) = D_{rGWN}(M_6)$ and derive γ by evaluating M_8/M_7.

Figure 3 illustrates that U_a impersonates S_j using γ, ID_{S_j}, s_j, and f_j and cheats the GWN and any other U during an authentication session. In Step 2 of the authenticated key agreement phase, U_a uses M_3/γ instead of M_3 mod k_j to recover r_{GWN}. In Step 4 of the authenticated key agreement phase, U_a uses M_8/γ instead of M_8 mod k_j to recover M_7. Other operations of U_a and S_j are exactly the same. After the authentication session, U shares $K_{U\text{-}Sj} = h(ID_U \| ID_{GWN} \| r_{GWN} \| r_U \| M_7 \| K_{GWN\text{-}U})$ with U_a instead of S_j and updates a new temporary identity $TID_U{}^{new}$.

3.2. Impersonation of Other Users

Assume that any other user U runs the login phase and authenticated key agreement phase. In Step 1 and Step 3 of the authenticated key agreement phase, U_a eavesdrops on S_j's message $<M_3, M_4, M_5, TS_2>$ from GWN and U's message $<M_{10}, M_{11}, M_{12}, TS_4>$ from GWN. From Section 3.1, we know that U_a obtains GWN's γ. Hence, U_a computes $r_{GWN} = M_3/\gamma$ and $(ID_U, ID_{GWN}, r_U, r_{GWN} \oplus K_{GWN\text{-}U}) = D_{rGWN}(M_4)$. Since U_a has U's ID_U and $K_{GWN\text{-}U}$, U_a can further compute U's new temporary identity $TID_U{}^{new}$ by computing $h(ID_U \| K_{GWN\text{-}U} \| TS_4) \oplus M_{11}$. Now, U_a can exploit U's ID_U, $K_{GWN\text{-}U}$, and $TID_U{}^{new}$ to impersonate U in a new authentication session.

3.3. Impersonation of GWN

U_a can impersonate GWN to cheat U and S_j. First, U_a obtains ID_U, $K_{GWN\text{-}U}$, ID_{GWN}, and γ as described in Section 3.2. Figure 4 shows how U_a impersonates GWN. In Step 3 of the authenticated key agreement phase, U_a neither decrypts M_6, retrieves $K_{GWN\text{-}S_j}$, nor computes $M_7 = h(K_{GWN\text{-}S_j} \| r_{GWN})$. Instead, U_a directly replaces M_7 with his/her random RN. Note that both U and S_j should authenticate each other and share $K_{U\text{-}Sj} = h(ID_U \| ID_{GWN} \| r_{GWN} \| r_U \| RN \| K_{GWN\text{-}U})$, because they do not check the validity of M_7.

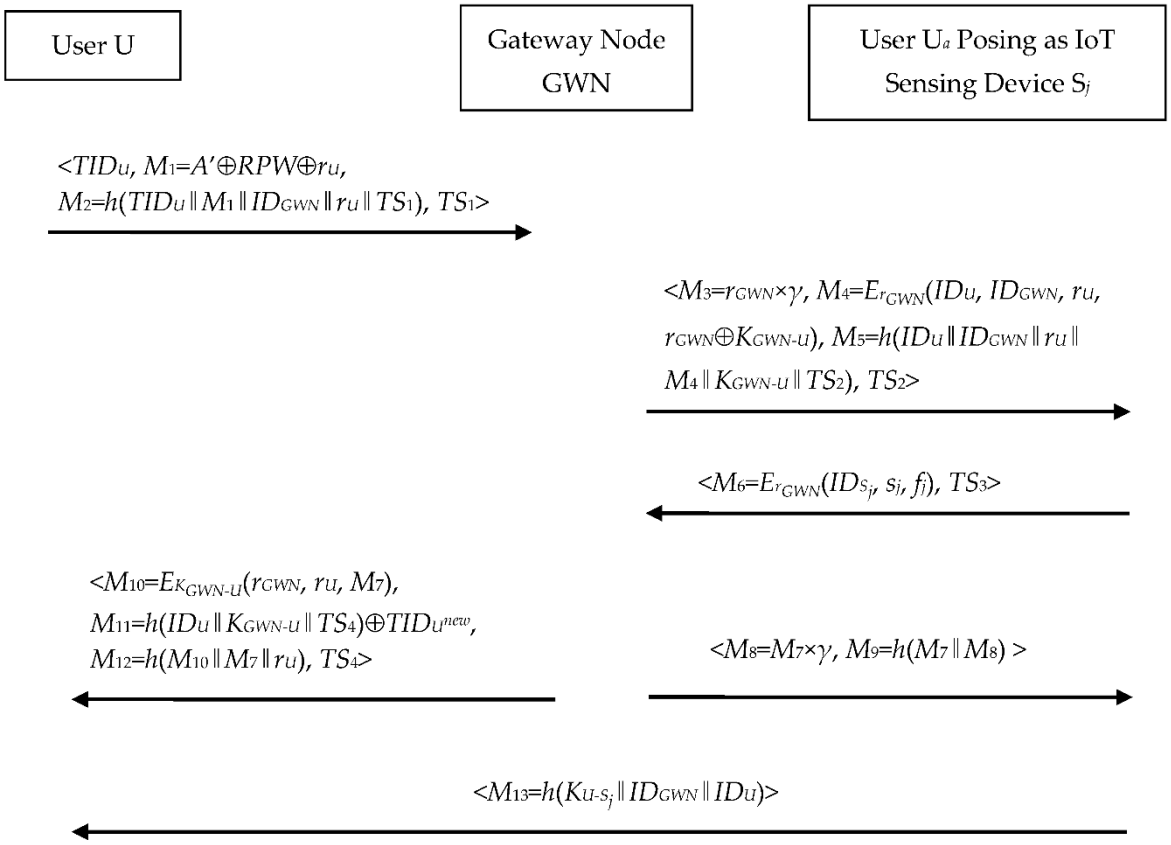

Figure 3. User impersonation attack on IoT sensing devices.

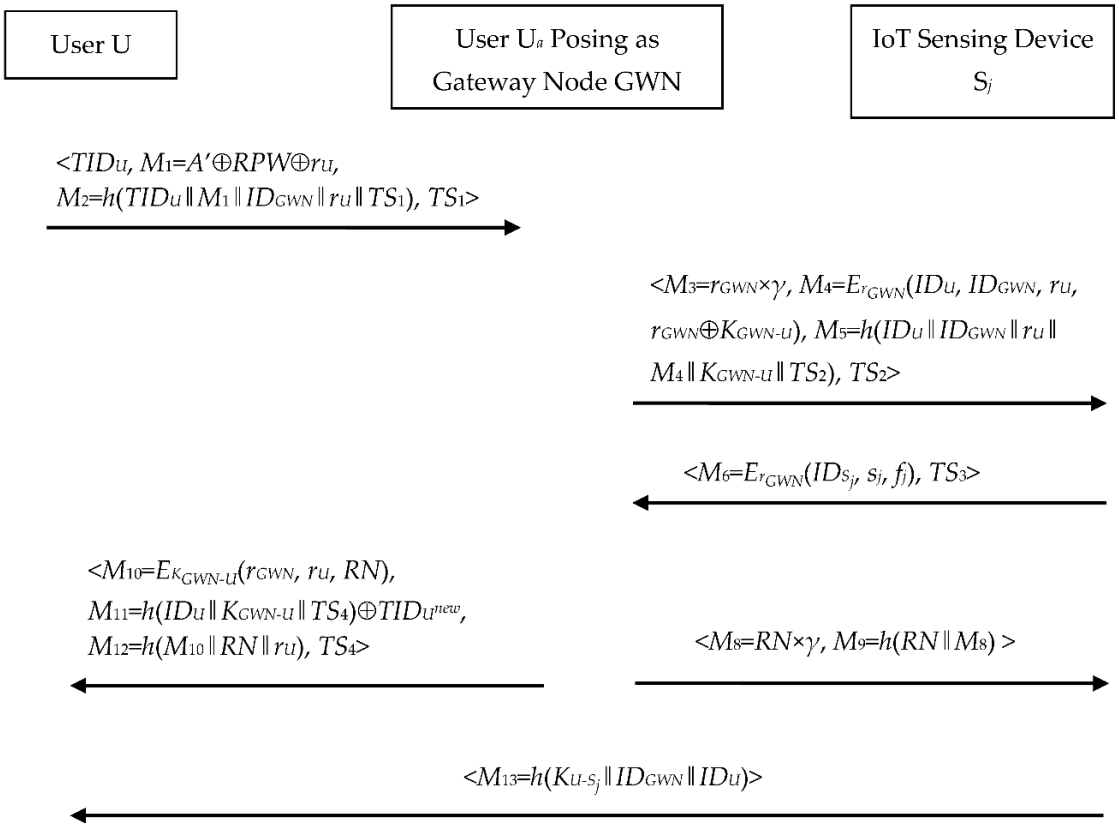

Figure 4. User impersonation attack on the GWN.

3.4. Further Disscussion

In every authentication session of Vinoth et al.'s scheme, the GWN uses its long-term secret key γ to secure its short-term secret key r_{GWN} for each user and each IoT sensing device. However, any user can directly recover γ after an authentication session. Hence, the user derives all the secrets of other users, the GWN, and IoT sensing devices and implements the impersonation attacks. To defeat a user's impersonation attack, γ cannot be disclosed to users.

User impersonation attacks are a serious threat under industrial IoT environments. Malicious users may impersonate other, honest users to collect sensitive industrial data or set dangerous processing instructions. By impersonating IoT sensing devices, malicious users can provide fake industrial data to other users. If malicious users employ impersonation of the GWN, they can manipulate a secure connection between the target user and IoT sensing devices. That is, malicious users can decide which IoT sensing devices can be connected to the target user.

4. IoT Sensing Device Impersonation Attacks

We showed that Vinoth et al.'s scheme was vulnerable to IoT sensing device impersonation attacks. That is, any legal but corrupt sensing device could impersonate users, the GWN, and any other IoT sensing devices in the deployed network. We assumed that S_j is a legal but corrupt IoT sensing device.

4.1. Impersonation of Users

To obtain TID_U, $S_j$eavesdrops on GWN's message <TID_U, M_1, M_2, TS_1> during Step 2 of the login phase. $S_j$further obtains U's ID_U, ID_{GWN}, and $K_{GWN\text{-}U}$ in Step 2 of the authenticated key agreement phase. However, S_j does not return the message <M_6, TS_3> to GWN. In this situation, both U and GWN terminate this session and therefore fail to update TID_U. Alternatively, S_j returns the message <M_6, TS_3> to GWN in Step 2 of the authenticated key agreement phase and further eavesdrops on U's message <M_{10}, M_{11}, M_{12}, TS_4> during Step 3 of the authenticated key agreement phase. At this time, S_j further obtains TID^{new} by computing $h(ID_U \| K_{GWN\text{-}U} \| TS_4) \oplus M_{13}$. Now, S_j knows all of U's secrets. As shown in Figure 5, S_j can start a new authentication session and perform the following steps to impersonate U:

(1) In Step 2 of login phase, $S_j$uses $K_{GWN\text{-}U}$, TID_U, and ID_{GWN} to generate M_1 and M_2.

(2) In Step 5 of authenticated key agreement phase, $S_j$does exactly the same as U.

At the end of the authentication session, S_m ($1 \leq m \neq j \leq n$) authenticates S_j as U and shares $K_{U\text{-}Sj} = h(ID_U \| ID_{GWN} \| r_{GWN} \| r_U \| M_7 \| K_{GWN\text{-}U})$ with S_j.

4.2. Impersonation of GWN

Moreover, in Step 2 of the authenticated key agreement phase, S_j can use k_j to compute $r_{GWN} \equiv M_3 \bmod k_j$. Hence, $S_j$can further derive GWN's γ by computing M_3/r_{GWN}. Now, S_j can exploit TID_U, ID_U, $K_{GWN\text{-}U}$, ID_{GWN}, and γ to impersonate GWN. As shown in Figure 6, the fake GWN impersonated by S_j omits M_6, generates its own RN, and then replaces M_7 with RN. Both U and S_j believe that RN is a legal M_7 because they do not check the validity of M_7. Finally, both U and S_m authenticate each other and share $K_{U\text{-}Sj} = h(ID_U \| ID_{GWN} \| r_{GWN} \| r_U \| RN \| K_{GWN\text{-}U})$.

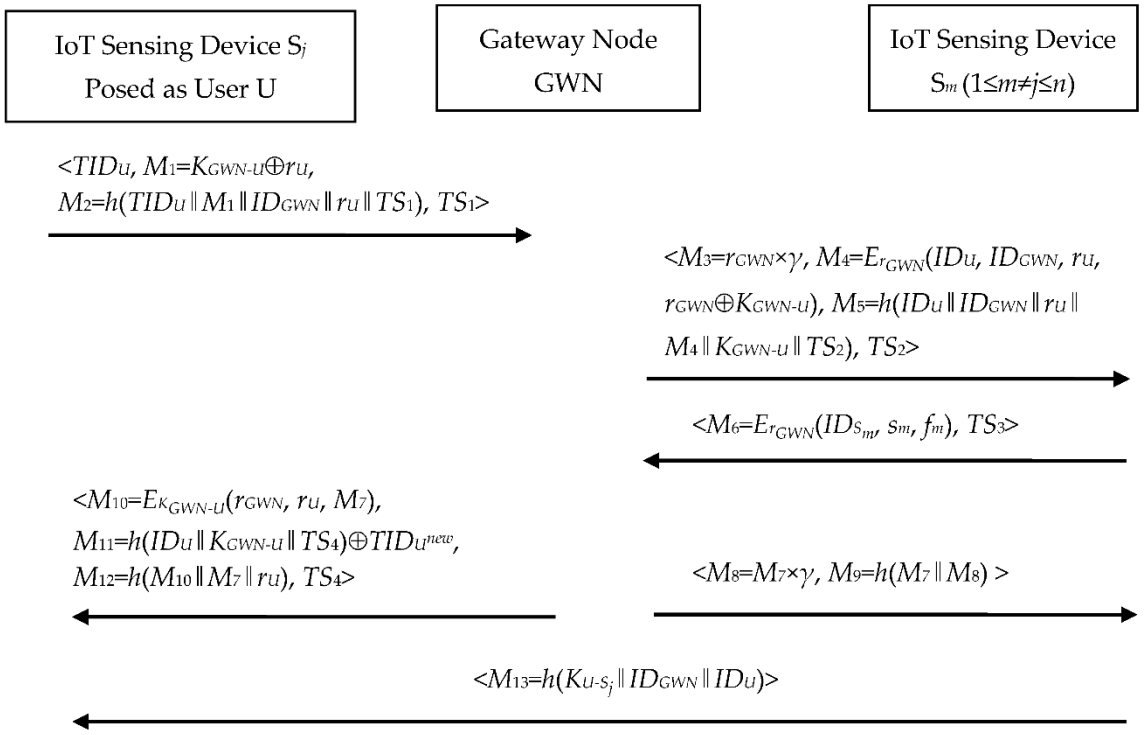

Figure 5. IoT sensing device impersonation attack on a user.

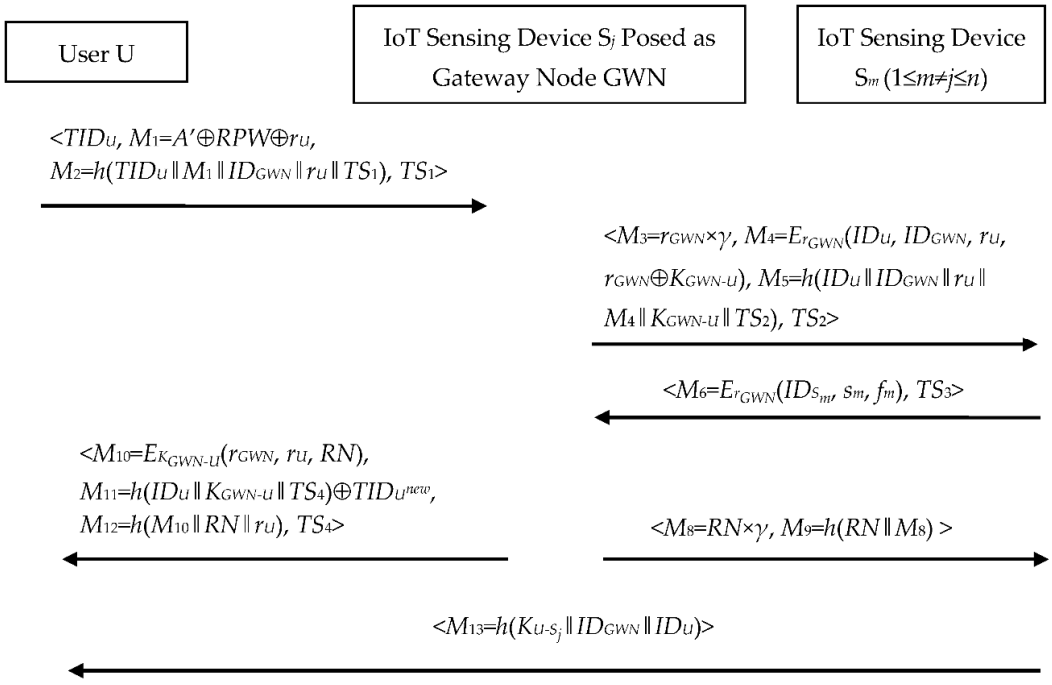

Figure 6. IoT sensing device impersonation attack on the GWN.

4.3. Impersonation of Other IoT Sensing Devices

If S_j wants to impersonate any other IoT sensing device $S_m (1 \le m \ne j \le n)$, S_j first eavesdrops on S_m's message <M_6, TS_3> during Step 2 of the authenticated key agreement phase and computes $(ID_{Sm}, s_m, f_m) = D_{rGWN}(M_6)$.

As shown in Figure 7, S_j can impersonate S_m using ID_{Sm}, s_m, f_m, and k_j in a new authentication session. In Step 2 of the authenticated key agreement phase, S_j recovers r_{GWN} by computing M_3 mod k_j. Then, S_j uses ID_{Sm}, s_m, f_m to fabricate S_m's M_6. In Step 4 of the authenticated key agreement phase, S_j calculates M_7 by computing M_8 mod

k_j. At the end of the new authentication session, U believes that S_j is S_m and shares $K_{U\text{-}Sj} = h(ID_U \| ID_{GWN} \| r_{GWN} \| r_U \| M_7 \| K_{GWN\text{-}U})$ with S_j.

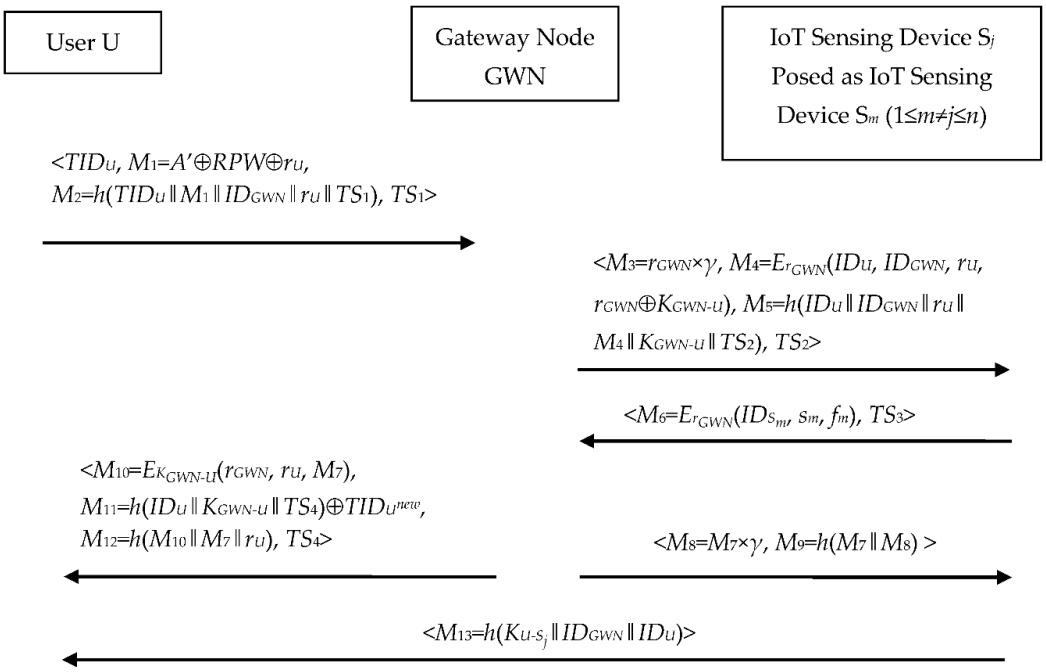

Figure 7. IoT sensing device impersonation attack on another IoT sensing device.

4.4. Further Disscussion

A legal but corrupt S_j can derive U's TID_U, ID_U, and $K_{GWN\text{-}U}$; GWN's ID_{GWN} and γ; and another IoT sensing device S_m's ID_{Sm}, s_m, and f_m from the public messages of the authentication session. Hence, S_j successfully impersonates U, GWN, and S_m by exploiting those secret parameters. To defeat the proposed attacks, Vinoth et al.'s scheme should avoid disclosing the secret parameters of other entities to S_j.

Industrial IoT sensing devices are perhaps exposed to hostile environments. An attacker may hijack and compromise industrial IoT sensing devices by physical means or Trojan horses. Once the attackers control an industrial IoT sensing device, they can subvert the industrial IoT sensing system just like the malicious user described in Section 3.4.

5. Replay Attack

As shown in Figure 2, we found that S_j's TS_3 in the message $<M_6, TS_3>$ was not protected by any cryptographic mechanism. Based on this observation of Vinoth et al.'s scheme, an outside attacker can eavesdrop on a valid message $<M_6, TS_3>$ in a normal run of the authenticated key agreement phase. Then, the attacker reuses M_6 and attaches the current timestamp TS_3* to impersonate S_j. Figure 8 describes this replay attack on Vinoth et al.'s scheme. After the replay attack, GWN believes that the attacker is S_j, although the attacker does not know any secret of S_j. Meanwhile, U does not authenticate the attacker as S_j. Note that GWN actually finishes its session in Step 3 of the authenticated key agreement phase and updates U's temporary identity. As a result, GWN updates the old TID_U to a new TID_U^{new}, but U still keeps the old TID_U. This means that U cannot log into the deployed network anymore, because during Step 1 of the authenticated key agreement phase, GWN fails to retrieve ID_U and $K_{GWN\text{-}U}$ according to U's old TID_U. For the industrial IoT sensing system, the legal user faces denial of service once the attacker implements the replay attack.

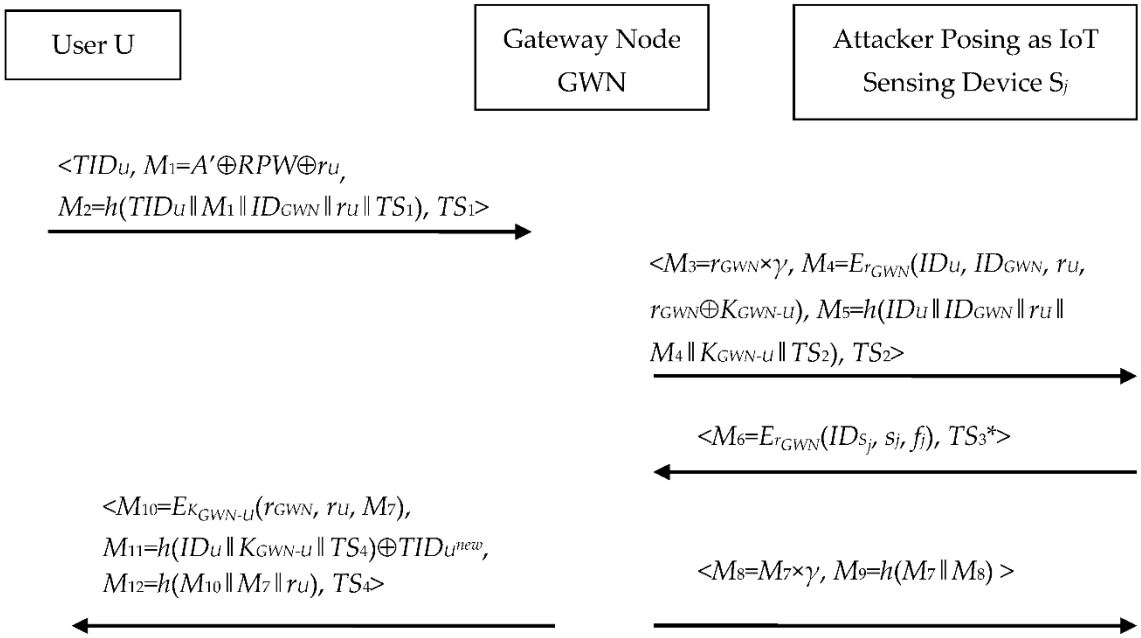

Figure 8. Replay attack.

To fix this vulnerability, we suggest that TS_3 should be protected by the cryptographic mechanism. For example, $S_j$could compute $M_6 = E_{rGWN}(ID_{S_j}, s_j, f_j, TS_3)$ instead of $M_6 = E_{rGWN}(ID_{S_j}, s_j, f_j)$ in Step 2 of the authenticated key agreement phase.

6. Desynchronization Attack

In Vinoth et al.'s scheme, U and GWN keep the same TID_U to authenticate each other. Hence, as shown in Figure 9, an outside attacker can block the message <M_{10}, M_{11}, M_{12}, TS_4> in Step 3 of the authenticated key agreement phase and instead send the message <M_{10}, RN, M_{12}, TS_4*> to U. Here, TS_4* is the attacker's current timestamp. During Step 5 of the authenticated key agreement phase, U confirms the freshness of TS_4* in the fabricated message <M_{10}, RN, M_{12}, TS_4*>, decrypts r_U from M_{10}, and successfully verifies r_U and M_{12}. U also computes $K_{U\text{-}Sj} = h(ID_U \| ID_{GWN} \| r_{GWN} \| r_U \| M_7 \| K_{GWN\text{-}U})$ and verifies M_{13}. Then, U further updates TID_U to $TID_U{}^{new} = h(ID_U \| K_{GWN\text{-}U} \| TS_4^*) \oplus RN$. The $TID_U{}^{new}$ computed by U is not equal to the $TID_U{}^{new}$ generated by GWN during Step 3 of the authenticated key agreement phase. This causes the failure to authenticate in the subsequent runs of Vinoth et al.'s scheme, though U, GWN, and $S_j$are all legal and honest.

The attacker randomly changes M_{11} because Vinoth et al.'s scheme does not check the authenticity of M_{11}. To overcome the desynchronization attack, our suggestion is to apply the message authentication code algorithm for M_{11}. Where industrial IoT sensing applications are concerned, this desynchronization attack has the same negative impact as the replay attack discussed in Section 5.

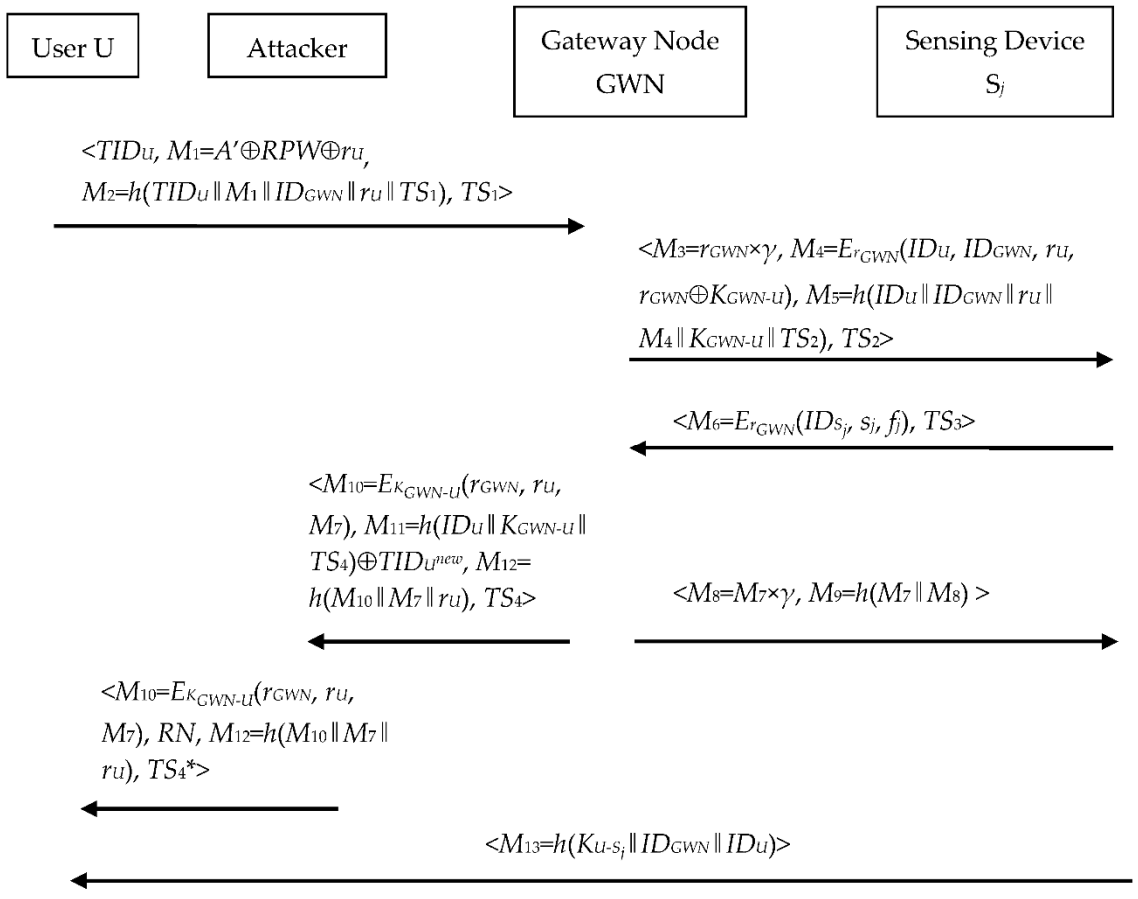

Figure 9. Desynchronization attack.

7. Weakness of User Privacy

In the authenticated key agreement scheme, user privacy guarantees that the attacker cannot derive the user's identity from the transmitted messages of the authentication sessions. This is called user anonymity. Moreover, the attacker also fails to link two different authentication sessions to the same user. This is called untraceability. User privacy is a concern in industrial IoT sensing applications, as users' private data can be leaked and misused if a factory deployed with IoT sensing devices is subjected to cyberattacks. For example, users' presence or absence at the industrial control room can be revealed simply by observing authentication sessions.

Vinoth et al. claimed that their scheme supported both user anonymity and untraceability because it employed the temporal TID_U to hide U's long-term ID_U. Furthermore, the symmetric encryption algorithm and cryptographic hash function were utilized to protect U's ID_U. In Section 3.2, we show that U_a can attain any other target user U's ID_U, $K_{GWN\text{-}U}$, and TID_U^{new}. Hence, when U_a finds TID_U^{new} in the message <TID_U, M_1, M_2, TS_1> during Step 2 of the login phase, U_a knows that U is running a session. Then, U_a can eavesdrop on the message <M_{10}, M_{11}, M_{12}, TS_4> in Step 3 of the authenticated key agreement phase and synchronously update U's new temporal TID_U^{new} by computing $h(ID_U \| K_{GWN\text{-}U} \| TS_4)\oplus M_{11}$. As a result, U_a can track any U all the time. S_j also can track any target user U. S_j first derives U's TID_U, ID_U, and $K_{GWN\text{-}U}$ as discussed in Section 4.1. Then, S_j uses TID_U to identify U, eavesdrops on the message <M_{10}, M_{11}, M_{12}, TS_4> during Step 3 of the authenticated key agreement phase and updates TID_U just like U. In conclusion, Vinoth et al.'s scheme fails to provide the user privacy protection.

Vinoth et al.'s scheme suffers from weak user privacy because it is vulnerable to user and IoT sensing device impersonation attacks. Vinoth et al.'s scheme could pro-

vide better user privacy if both user and IoT sensing device impersonation attacks are repaired correctly.

8. Conclusions and Future Work

In Vinoth et al.'s scheme, the user and multiple IoT sensing devices negotiate a secret session key via a group key, i.e., r_{GWN}. This novel design improves the efficiency of Vinoth et al.'s scheme. It is a desirable feature of the IoT sensing applications. Hence, we study Vinoth et al.'s scheme in aspects of security and privacy. Although Vinoth et al.'s scheme proved secure under the Canetti–Krawczyk threat model [40], we still revealed several serious security and privacy vulnerabilities in the scheme. In addition, Vinoth et al.'s scheme employs random numbers such as r_U and r_{GWN} and timestamps such as TS_1, TS_2, TS_3, and TS_4 at the same time. It is widely known that random numbers and timestamps are both used to defeat reply attacks and ensure the freshness of the message. From the perspective of applications, the use of both random numbers and timestamps increases the complexity of the authentication system and brings greater security risk. Therefore, it would be best to adopt only one of them in an authenticated key agreement scheme.

It is still a challenge to design a robust and efficient authenticated key agreement scheme for IoT sensing applications. One avenue for future work is to formulate a communication model appropriate for defining authentication and key agreement goals and present the definitions of security and privacy under the communication model. The results of our analysis of Vinoth et al.'s scheme can provide a reference for these definitions. Another avenue for future work is to develop an authenticated key agreement scheme that not only satisfies our formal definitions but also achieves high efficiency. In [41], Bellare and Rogaway proposed a security definition, a protocol, and a proof for secure session key distribution with the trust three-party case. One feasible idea is to extend Bellare and Rogaway's definition and protocol for IoT sensing models. We expect that this will require a great deal of research work to accomplish.

Funding: The work of Da-Zhi Sun was supported in part by the National Natural Science Foundation of China under Grant No. 61872264. The APC was funded by the National Natural Science Foundation of China under Grant No. 61872264.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The author would like to thank the editor and the reviewers for their valuable suggestions and comments.

Conflicts of Interest: The author declares no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Hirman, M.; Benesova, A.; Sima, K.; Steiner, F.; Tupa, J. Design, fabrication and risk assessment of IoT unit for products manufactured in industry 4.0 factory. *Procedia Manuf.* **2020**, *51*, 1178–1183. [CrossRef]
2. Macioszek, E.; Kurek, A. Extracting road traffic volume in the city before and during COVID-19 through video remote sensing. *Remote Sens.* **2021**, *13*, 2329. [CrossRef]
3. Hassija, V.; Chamola, V.; Saxena, V.; Jain, D.; Goyal, P.; Sikdar, B. A survey on IoT security: Application areas, security threats, and solution architectures. *IEEE Access* **2019**, *7*, 82721–82743. [CrossRef]
4. Vinoth, R.; Deborah, L.J.; Vijayakumar, P.; Kumar, N. Secure multifactor authenticated key agreement scheme for industrial IoT. *IEEE Internet Things J.* **2021**, *8*, 288–296. [CrossRef]
5. Kumari, S.; Khan, M.K.; Atiquzzaman, M. User authentication schemes for wireless sensor networks: A review. *Ad Hoc Netw.* **2015**, *27*, 159–194. [CrossRef]
6. Singh, D.; Kumar, B.; Singh, S.; Chand, S. Evaluating authentication schemes for real-time data in wireless sensor network. *Wirel. Pers. Commun.* **2020**, *114*, 629–655. [CrossRef]
7. Sun, D.Z.; Li, J.X.; Feng, Z.Y.; Cao, Z.F.; Xu, G.Q. On the security and improvement of a two-factor user authentication scheme in wireless sensor networks. *Pers. Ubiquitous Comput.* **2013**, *17*, 895–905. [CrossRef]

8. Wang, D.; Wang, P. Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks. *Ad Hoc Netw.* **2014**, *20*, 1–15. [CrossRef]
9. Jiang, Q.; Ma, J.; Lu, X.; Tian, Y.L. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. *Peer-Peer Netw. Appl.* **2015**, *8*, 1070–1081. [CrossRef]
10. Wei, F.; Zhang, R.; Shen, J. A Provably Secure Two-Factor Authenticated Key Exchange Protocol for Wireless Sensor Networks Based on Authenticated Encryption. In *Lecture Notes on Data Engineering and Communications Technologies, Proceedings of the 11th International Conference on Advances on Broad-Band Wireless Computing, Communication and Applications (BWCCA 2016), Asan, Korea, 5–7 November 2016*; Barolli, L., Xhafa, F., Yim, K., Eds.; Springer: Cham, Switzerland, 2017; Volume 2, pp. 849–855.
11. Wu, F.; Xu, L.L.; Kumari, S.; Li, X. A new and secure authentication scheme for wireless sensor networks with formal proof. *Peer-Peer Netw. Appl.* **2017**, *10*, 16–30. [CrossRef]
12. Wu, F.; Li, X.; Sangaiah, A.K.; Xu, L.L.; Kumari, S.; Wu, L.X.; Shen, J. A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks. *Future Gener. Comput. Syst.* **2018**, *82*, 727–737. [CrossRef]
13. Chandrakar, P. A secure remote user authentication protocol for healthcare monitoring using wireless medical sensor networks. *Int. J. Ambient Comput. Intell.* **2019**, *10*, 6. [CrossRef]
14. Kaur, D.; Kumar, D. Cryptanalysis and improvement of a two-factor user authentication scheme for smart home. *J. Inf. Secur. Appl.* **2021**, *58*, 102787.
15. Qi, M.P.; Chen, J.H. Secure authenticated key exchange for WSNs in IoT applications. *J. Supercomput.* **2021**. [CrossRef]
16. Das, A.K. An efficient and novel three-factor user authentication scheme for large-scale heterogeneous wireless sensor networks. *Int. J. Commun. Netw. Distrib. Syst.* **2015**, *15*, 22–60. [CrossRef]
17. Das, A.K. A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks. *Peer-Peer Netw. Appl.* **2016**, *9*, 223–244. [CrossRef]
18. Wang, C.Y.; Xu, G.A.; Sun, J. An enhanced three-factor user authentication scheme using elliptic curve cryptosystem for wireless sensor networks. *Sensors* **2017**, *17*, 2946. [CrossRef] [PubMed]
19. Wu, F.; Xu, L.L.; Kumari, S.; Li, X. An improved and provably secure three-factor user authentication scheme for wireless sensor networks. *Peer-Peer Netw. Appl.* **2018**, *11*, 1–20. [CrossRef]
20. Shin, S.; Kwon, T. A lightweight three-factor authentication and key agreement scheme in wireless sensor networks for smart homes. *Sensors* **2019**, *19*, 2012. [CrossRef]
21. Luo, H.G.; Wen, G.J.; Su, J. Lightweight three factor scheme for real-time data access in wireless sensor networks. *Wirel. Netw.* **2020**, *26*, 955–970. [CrossRef]
22. Jabbari, A.; Mohasef, J.B. Improvement of a user authentication scheme for wireless sensor networks based on internet of things security. *Wirel. Pers. Commun.* **2021**, *116*, 2565–2591. [CrossRef]
23. Jiang, Q.; Kumar, N.; Ma, J.F.; Shen, J.; He, D.B.; Chilamkurti, N. A privacy-aware two-factor authentication protocol based on elliptic curve cryptography for wireless sensor networks. *Int. J. Netw. Manag.* **2017**, *27*, e1937. [CrossRef]
24. Adavoudi-Jolfaei, A.; Ashouri-Talouki, M.; Aghili, S.F. Lightweight and anonymous three-factor authentication and access control scheme for real-time applications in wireless sensor networks. *Peer-Peer Netw. Appl.* **2019**, *12*, 43–59. [CrossRef]
25. Lu, Y.R.; Xu, G.Q.; Li, L.X.; Yang, Y.X. Anonymous three-factor authenticated key agreement for wireless sensor networks. *Wirel. Netw.* **2019**, *25*, 1461–1475. [CrossRef]
26. Sadri, M.J.; Asaar, M.R. A lightweight anonymous two-factor authentication protocol for wireless sensor networks in internet of vehicles. *Int. J. Commun. Syst.* **2020**, *33*, e4511. [CrossRef]
27. Far, H.A.N.; Bayat, M.; Das, A.K.; Fotouhi, M.; Pournaghi, S.M.; Doostari, M.A. LAPTAS: Lightweight anonymous privacy-preserving three-factor authentication scheme for WSN-based IIoT. *Wirel. Netw.* **2021**, *27*, 1389–1412.
28. Das, A.K.; Sutrala, A.K.; Kumari, S.; Odelu, V.; Wazid, M.; Li, X. An efficient multi-gateway-based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks. *Secur. Commun. Netw.* **2016**, *9*, 2070–2092. [CrossRef]
29. Amin, R.; Biswas, G.P. A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. *Ad Hoc Netw.* **2016**, *36*, 58–80. [CrossRef]
30. Wu, F.; Xu, L.L.; Kumari, S.; Li, X.; Shen, J.; Choo, K.K.R.; Wazid, M.; Das, A.K. An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment. *J. Netw. Comput. Appl.* **2017**, *89*, 72–85. [CrossRef]
31. Sutrala, A.K.; Das, A.K.; Reddy, A.G.; Vasilakos, A.V.; Rodrigues, J.J.P.C. On the design of secure user authenticated key management scheme for multigateway-based wireless sensor networks using ECC. *Int. J. Commun. Syst.* **2018**, *31*, e3514. [CrossRef]
32. Guo, H.; Gao, Y.; Xu, T.G.; Zhang, X.Y.; Ye, J.F. A secure and efficient three-factor multi-gateway authentication protocol for wireless sensor networks. *Ad Hoc Netw.* **2019**, *95*, 101965. [CrossRef]
33. Lee, J.; Yu, S.; Park, K.; Park, Y.; Park, Y. Secure three-factor authentication protocol for multi-gateway IoT environments. *Sensors* **2019**, *19*, 2358. [CrossRef] [PubMed]
34. Xu, L.L.; Wu, F. A lightweight authentication scheme for multi-gateway wireless sensor networks under IoT conception. *Arab. J. Sci. Eng.* **2019**, *44*, 3977–3993. [CrossRef]
35. Wang, D.; Hong, S.H.; Wang, Q.X. Revisiting a multifactor authentication scheme in industrial IoT. *Secur. Commun. Netw.* **2021**, *2021*, 9995832. [CrossRef]

36. Vinoth, R.; Deborah, L.J. An efficient key agreement and authentication protocol for secure communication in industrial IoT applications. *J. Ambient Intell. Humaniz. Comput.* **2021**. [CrossRef]
37. Gupta, M.; Chaudhari, N.S. Anonymous two factor authentication protocol for roaming service in global mobility network with security beyond traditional limit. *Ad Hoc Netw.* **2019**, *84*, 56–67. [CrossRef]
38. Wang, F.F.; Xu, G.A.; Gu, L.Z. A secure and efficient ECC based anonymous authentication protocol. *Secur. Commun. Netw.* **2019**, *2019*, 4656281. [CrossRef]
39. Jiang, Q.; Zhang, N.; Ni, J.B.; Ma, J.F.; Ma, X.D.; Choo, K.K.R. Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles. *IEEE Trans. Veh. Technol.* **2020**, *69*, 9390–9401. [CrossRef]
40. Canetti, R.; Krawczyk, H. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In *Lecture Notes in Computer Science, Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2001), Innsbruck, Austria, 6–10 May 2001*; Pfitzmann, B., Ed.; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2045, pp. 453–474.
41. Bellare, M.; Rogaway, P. Provably Secure Session Key Distribution—The Three Party Case. In Proceedings of the 27th ACM Symposium on the Theory of Computing (STOC'95), Las Vegas, NV, USA, 29 May–1 June 1995; ACM: New York, NY, USA, 1995; pp. 57–66.